Assignment 06
(due on 12/24 19:00)

PS6_1.R

1. Matrix multiplication

1.1 [5 points] Write a subroutine `Matrix_multip.f90` to do matrix multiplication.

Answer:

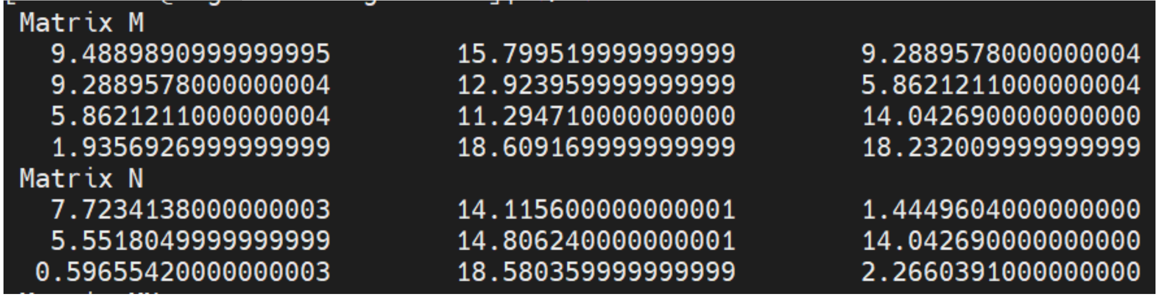
```
subroutine Matrix_multip(M,N,MN)
implicit none

        real(8),intent(in)::M(4,3),N(3,3)
        real(8),intent(out)::MN(4,3)
        real(8)::Mtemp
        INTEGER:: i,j,k
        ![a,k]=size(M)
        ![k,b]=size(N)
        do i=1,4
          do j=1,3
            Mtemp=0
            do k=1,3
                  Mtemp=Mtemp+M(i,k)*N(k,j)
            enddo
          MN(i,j)=Mtemp
          enddo
        enddo

                           print*,'Matrix MN'
        do i=1,4
          print*,MN(i,:)
        enddo
end subroutine Matrix_multip
```

1.2 [5 points] Write a program `Main.f90` to read `/work/ese-ouycc/fortran_2/M.dat` as the matrix `M`, and `/work/ese-ouycc/fortran_2/N.dat` as the matrix `N`.

Answer:

```
! Open the M file
open(unit=u, file='M.dat', status='old')
! Read data line by line and pass the value to M
read(u,*) M
! Close the file
close(u)
! Display the values
Print*, 'Matrix M'
do i = 1,4
  write(*,*) M(:,i)
```

```
enddo

! Open the N file
open(unit=u, file='N.dat', status='old')
! Read data line by line and pass the value to M
read(u,*) N
! Close the file
close(u)
! Display the values
Print*, 'Matrix N'
do i = 1,3
  write(*,*) N(:,i)
enddo

Mtran = transpose(M)
Ntran = transpose(N)
```

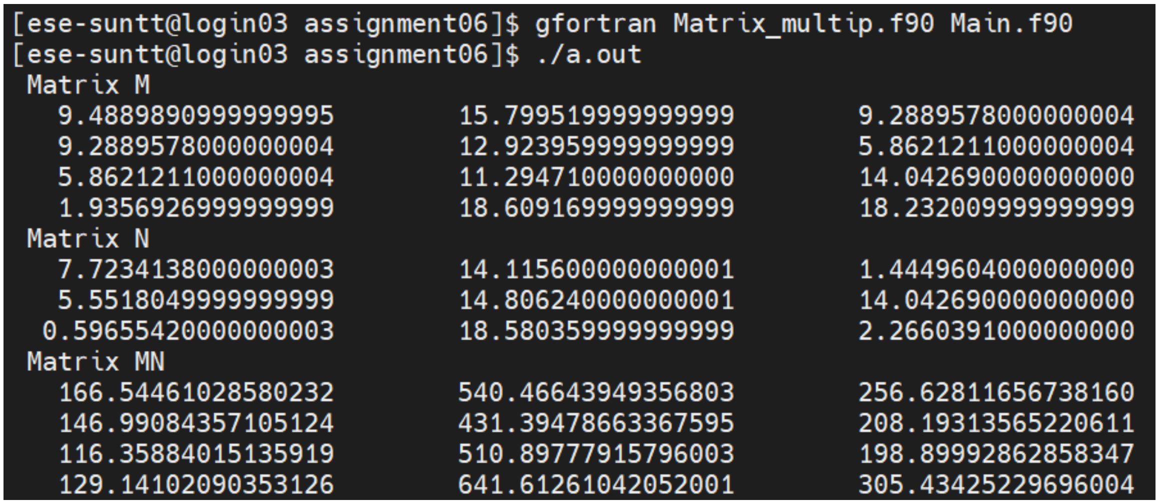


```
 Matrix M
   9.4889890999999995        15.799519999999999        9.2889578000000004
   9.2889578000000004        12.923959999999999        5.8621211000000004
   5.8621211000000004        11.294710000000000        14.042690000000000
   1.9356926999999999        18.609169999999999        18.232009999999999
 Matrix N
   7.7234138000000003        14.115600000000001        1.4449604000000000
   5.5518049999999999        14.806240000000001        14.042690000000000
  0.59655420000000003        18.580359999999999        2.2660391000000000
```

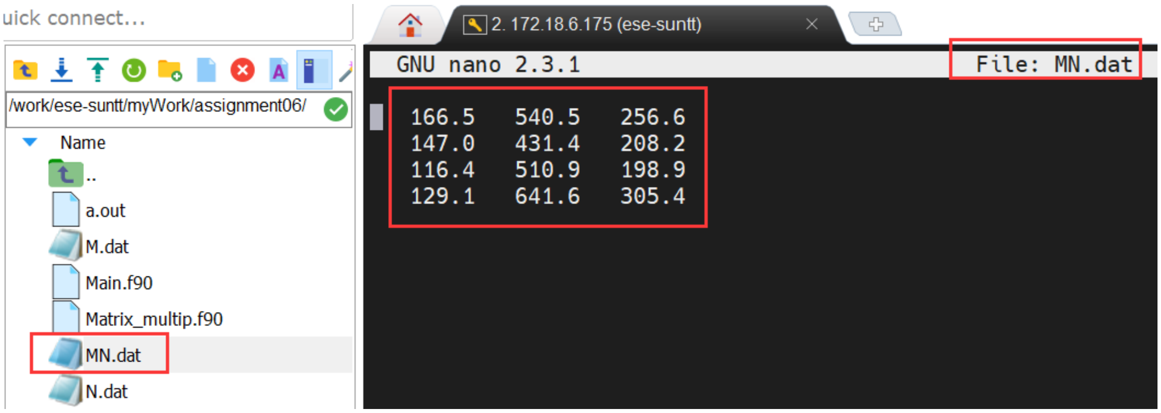
1.3 [5 points] Call subroutine `Matrix_multip()` from `Main.f90` to compute `M*N`; write the output to a new file `MN.dat`, values are in formats of `f8.1`.

Answer:

```
call Matrix_multip(Mtran,Ntran,MN)

! Write the values to a new file, in a certain format
MNtran = transpose(MN)
open(unit=u, file='MN.dat', status='replace')
do i = 1,4
   write(u, '(f8.1,f8.1,f8.1)') MNtran(:,i)
enddo
close(u)
```



```
[ese-suntt@login03 assignment06]$ gfortran Matrix_multip.f90 Main.f90
[ese-suntt@login03 assignment06]$ ./a.out
 Matrix M
   9.4889890999999995        15.799519999999999        9.2889578000000004
   9.2889578000000004        12.923959999999999        5.8621211000000004
   5.8621211000000004        11.294710000000000        14.042690000000000
   1.9356926999999999        18.609169999999999        18.232009999999999
 Matrix N
   7.7234138000000003        14.115600000000001        1.4449604000000000
   5.5518049999999999        14.806240000000001        14.042690000000000
  0.59655420000000003        18.580359999999999        2.2660391000000000
 Matrix MN
   166.54461028580232        540.46643949356803        256.62811656738160
   146.99084357105124        431.39478663367595        208.19313565220611
   116.35884015135919        510.89777915796003        198.89992862858347
   129.14102090353126        641.61261042052001        305.43425229696004
```



```
 GNU nano 2.3.1                                              File: MN.dat

 166.5    540.5    256.6
 147.0    431.4    208.2
 116.4    510.9    198.9
 129.1    641.6    305.4
```

PS6_2.R

2. Calculate the solar zenith angle

The Solar Zenith Angle (SZA; θz in the following figure [1]) is the angle between the local zenith and the line of sight from that point to the Sun. This means that the higher the Sun is in the sky, to lower the SZA is. The value of the SZA depends on the location on the Earth and the local date and time.

Please read this note (page 1-2) for more about how to calculate SZA.

2.1 [5 points] Write a module `Declination_angle` to calculate the *declination angle* on a certain date.

[**Hint:** using equation 2]

Answer:

```
module Declination_angle

implicit none

  !real, parameter :: pi = 3.1415926536
  !integer :: N
```

```
contains

  subroutine Decl_angle(N,A)
          implicit none
          integer, intent(in)  :: N
   real, intent(out)    :: A
          real                                    :: pi
          pi = 3.1415926536
          A=23.45*sin(((N+284.)*360/365)*pi/180)
   print*, "A is", A
  end subroutine Decl_angle

end module Declination_angle
```

2.2 [10 points] Write a module `AST` to calculate the *apparent solar time* (AST; or *local solar time*) in a certain location for a certain date and time.

[**Hint:** using equation 3-5]

Answer:

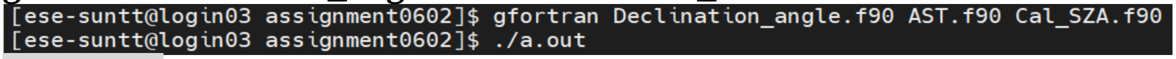
```
module AST

implicit none

  real, parameter :: pi = 3.1415926536
  !integer                              :: N

contains

  subroutine A_solar_time(N,Long,LST,ASTcal)
  implicit none
  integer, intent(in)  :: N
  real, intent(in)  :: Long,LST
  integer, intent(out) :: ASTcal
  real                             :: D,ET,LSTM
          D = (360*(N-81.)/365)*pi/180
          ET = 9.87*sin(2*D) - 7.53*cos(D) - 1.5*sin(D)
          LSTM = 15*(nint(Long/15))
          ASTcal = LST + 4*(LSTM - Long) + ET
   print*, "AST is ", ASTcal
  end subroutine A_solar_time

end module AST
```

2.3 [10 points] Write a main program (`Cal_SZA.f90`) that uses module `Declination_angle` and `AST` to print the SZA in a certain location for a certain date and time.

[**Hint:** using equation 6-7]

Answer:

gfortran Declination_angle.f90 AST.f90 Cal_SZA.f90



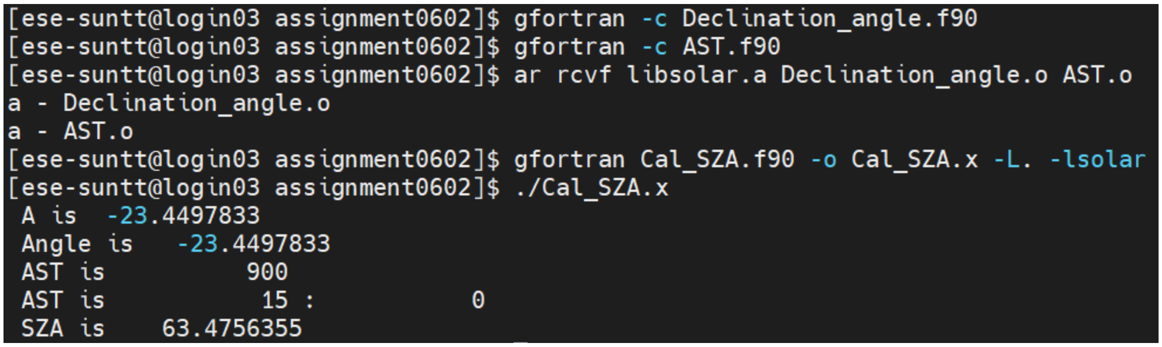
```
[ese-suntt@login03 assignment0602]$ gfortran Declination_angle.f90 AST.f90 Cal_SZA.f90
[ese-suntt@login03 assignment0602]$ ./a.out
```

```
program Main

use AST
use Declination_angle

implicit none

!real(4), parameter :: pi = 3.1415926536
real                                    :: A, Long, LST, H, Lat, SZA
integer :: N, ASTcal


!2.1Calculate the declination angle on 12.20 of Shenzhen.
N=355
call Decl_angle(N,A)
write(*,*),"Angle is ",A

!2.2Calculate the AST in Shenzhen (22.542883N, 114.062996E) for 14:35 on 2020-12-20.
Long = 114.062996
!LST = 14:35,which change to minute is 14*60+35=875min
LST = 875
call A_solar_time(N,Long,LST,ASTcal)
!Change minute to Hour:minute
write(*,*),"AST is ",floor(ASTcal/60.),":",mod(ASTcal,60)

!2.3Print the SZA in a certain location for a certain date and time.
H = (ASTcal -720)/4
Lat = 22.542883
SZA = ACOS(cos(Lat*pi/180)*cos(A*pi/180)*cos(H*pi/180) +sin(Lat*pi/180)*sin(A*pi/180))
*180/pi
write(*,*),"SZA is ",SZA

end program Main
```

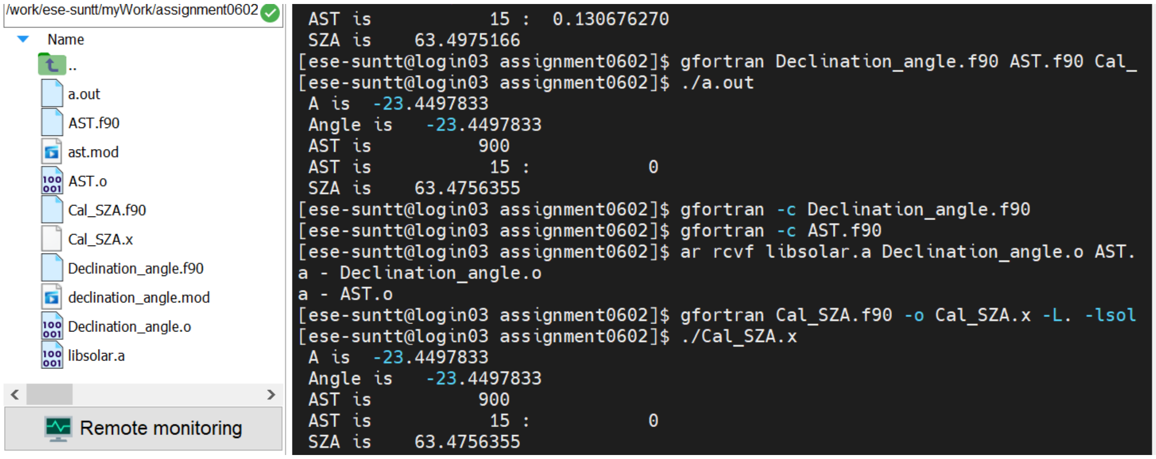
2.4 [5 points] Create a library (`libsolar.a`) that contains `Declination_angle.o` and `AST.o`. Compile `Cal_SZA.f90` using `libsolar.a`.

Answer:

gfortran -c Declination_angle.f90
gfortran -c AST.f90
ar rcvf libsolar.a Declination_angle.o AST.o
gfortran Cal_SZA.f90 -o Cal_SZA.x -L. -lsolar



```
[ese-suntt@login03 assignment0602]$ gfortran -c Declination_angle.f90
[ese-suntt@login03 assignment0602]$ gfortran -c AST.f90
[ese-suntt@login03 assignment0602]$ ar rcvf libsolar.a Declination_angle.o AST.o
a - Declination_angle.o
a - AST.o
[ese-suntt@login03 assignment0602]$ gfortran Cal_SZA.f90 -o Cal_SZA.x -L. -lsolar
[ese-suntt@login03 assignment0602]$ ./Cal_SZA.x
 A is  -23.4497833
 Angle is   -23.4497833
 AST is          900
 AST is           15 :           0
 SZA is    63.4756355
```

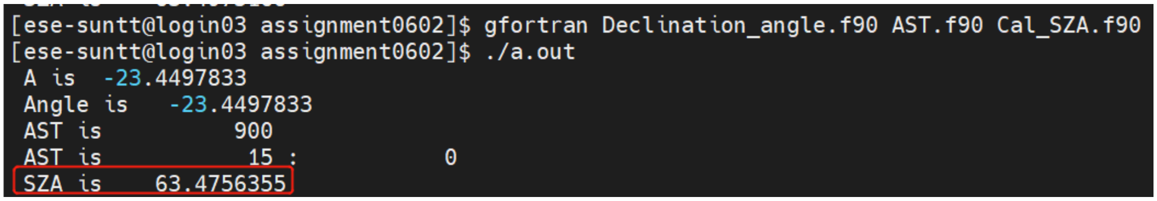


```
 AST is           15 :  0.130676270
 SZA is    63.4975166
[ese-suntt@login03 assignment0602]$ gfortran Declination_angle.f90 AST.f90 Cal_
[ese-suntt@login03 assignment0602]$ ./a.out
 A is  -23.4497833
 Angle is   -23.4497833
 AST is          900
 AST is           15 :           0
 SZA is    63.4756355
[ese-suntt@login03 assignment0602]$ gfortran -c Declination_angle.f90
[ese-suntt@login03 assignment0602]$ gfortran -c AST.f90
[ese-suntt@login03 assignment0602]$ ar rcvf libsolar.a Declination_angle.o AST.
a - Declination_angle.o
a - AST.o
[ese-suntt@login03 assignment0602]$ gfortran Cal_SZA.f90 -o Cal_SZA.x -L. -lsol
[ese-suntt@login03 assignment0602]$ ./Cal_SZA.x
 A is  -23.4497833
 Angle is   -23.4497833
 AST is          900
 AST is           15 :           0
 SZA is    63.4756355
```

2.5 [5 points] Print the SZA for Shenzhen (`22.542883N, 114.062996E`) at `14:35` (local time) on `2020-12-20`.

Answer:

The SZA is about 63.48 °.



```
[ese-suntt@login03 assignment0602]$ gfortran Declination_angle.f90 AST.f90 Cal_SZA.f90
[ese-suntt@login03 assignment0602]$ ./a.out
 A is  -23.4497833
 Angle is   -23.4497833
 AST is          900
 AST is           15 :           0
 SZA is    63.4756355
```